REMARKS

Claims 1, 6-10, 12, 14-17, and 19-20 are amended, and claim 2 is canceled herein. Claims 1 and 3-20 remain pending in the captioned case. Further examination and reconsideration of the presently claimed application are respectfully requested.

Objection to the Claims

An objection was lodged against claims 1-15 and 17. In response thereto, amendments are made to claims 1, 6, 8, 14, and 17 in a manner believed to obviate the objections in their entirety. Accordingly, removal of the objections to claims 1-15 and 17 is respectfully requested.

Section 112 Rejection

Claims 1-20 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. In response thereto, claims 1, 7-10, 12, and 14-17 are amended in a manner believe to obviate this rejection in its entirety. Accordingly, removal of the this rejection of claims 1-20 is respectfully requested.

Section 101 Rejection

Claims 1-15 were rejected under 35 U.S.C. § 101 for including non-statutory subject matter. To expedite prosecution, claims 1 and 8 are amended as suggested by the Examiner. Accordingly, Applicants respectfully request removal of this rejection in its entirety.

Section 102 Rejection

Claims 1-2 and 4-7 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2003/0110476 to Aihara (hereinafter "Aihara"). The standard for "anticipation" is one of fairly strict identity. A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art of reference. *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); MPEP 2131. Furthermore, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, as arranged in the claim. *W.L. Gore & Assocs. V. Garlock*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983). Using these standards, Applicants submit the cited art fails to disclose each and every element of the currently pending claims, some distinctive features of which are set forth in more detail below.

Aihara fails to anticipate a graphical user interface (GUI) capable of receiving user input to select an instruction address displayed on the screen, wherein upon selection, the screen/GUI displays a designator for instructions addresses that will proceed to a subsequent pipeline stage and a non-designator for instruction addresses that will not proceed (i.e., stall) to the subsequent pipeline stage. As noted above, claim 1 has been amended to clarify that a graphical user interface (GUI) is displayed on the display screen. Claim 1 has also been amended to clarify that the GUI is adapted to receive user input for selecting at least one of the instruction addresses displayed on the screen/GUI. Upon receiving user input within the GUI (e.g., upon receiving a mouse click at breakpoint field 66, or some other field, within GUI 64), the screen/GUI will display designators/non-designators for those instruction(s) that will proceed/not proceed to a subsequent stage of the pipeline. Support for the incorporation of a GUI into claim 1 may be found in originally filed claim 2, and Figs. 5 and 6 and corresponding sections of the text, e.g., pages 10-12 of the specification. As such, the amendments made to claim 1 do not introduce new matter.

The Office Action alleges that Aihara provides teaching for a graphical user interface (GUI) on page 5, ¶¶ 0062 and 0073 of Aihara. Applicants disagree. Aihara discloses a source code debugger and method for debugging a program (Aihara -- Title). An embodiment of the source code debugger is shown in Fig. 11, while an embodiment of the method is shown in Fig. 12 of Aihara. In Figs. 13-15, Aihara discloses a screen which is displayed on a display device (e.g., device 13 of Fig. 11) for displaying source code. As shown in Fig. 13A and described in corresponding portions of the text of Aihara, the source code displayed on the screen includes instruction addresses (e.g., ox800201ec), as well as instruction codes (e.g., LD $1, 0($12)) and current pipeline stages (e.g., W) attributed to the instruction addresses. Similar screen shots are shown in Figs. 13B, 14A, 14B, and 15 of Aihara.

Contrary to the allegations made in the Office Action, the screen shots disclosed by Aihara cannot be considered equivalent to a graphical user interface or GUI. As indicated above and throughout the teachings of Aihara, the screen shots are only capable of displaying the source code information. A screen which is only capable of displaying information is not equivalent to a GUI, which provides an interface by which a user may interact with the contents of the display screen. The teachings of Aihara provide absolutely no indication or means by which a user may interact with the information displayed on the screen. In particular, Aihara fails to teach or indicate that the screen shots (i.e., the alleged "GUI") may be capable of receiving user input as presently claimed.

In addition, Aihara fails to teach or indicate that the screen shots may display designators/non-designators for the instruction addresses that will proceed/not-proceed during times when user input is supplied to the screen shots. In fact, Aihara specifically teaches that the instruction codes, instruction addresses and stage information (e.g., whether an instruction address will proceed/stall within a particular stage) are automatically obtained and displayed once "ISS controlling module 14 halts execution [of] the cycle-accurate ISS 10" (Aihara -- ¶¶ 0049-0053; Fig. 12). No user interaction or input is needed to track the current position of the program processing.

Aihara fails to provide teaching or suggestion for all limitations of independent claim 1 by failing to disclose a graphical user interface (GUI) that is capable of receiving user input and displaying designators/non-designators in response to such input.

In addition to claim 1, Aihara fails to provide teaching or suggestion for many of the limitations recited in claims dependent therefrom. As but one example, Aihara fails to disclose "wherein the user actuates a pointing device to supply the user input to the GUI for selecting only one of the instruction addresses, wherein in response to said selection, the screen displays the designator over a field bearing a stage name for all of the instruction addresses that will proceed to the next stage in the processor pipeline," as recited in present claim 7.

For at least the reasons noted above, Applicant asserts that independent claim 1, as well as claims dependent therefrom, are not anticipated by the cited art. Accordingly, Applicants respectfully request removal of this rejection in its entirety.

Section 103 Rejection

Claims 8-10 and 12-18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Aihara in view of U.S. Patent No. 5,913,052 to Beatty et al. (hereinafter "Beatty"). Claims 3 and 11 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Aihara, Beatty, and U.S. Patent Application Publication No. 2002/0130871 to Hill (hereinafter "Hill"). To establish a case of *prima facie* obviousness of a claimed invention, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. Second, there must be a reasonable expectation of success. As stated in MPEP 2143.01, the fact that references can be hypothetically combined or modified is not sufficient to establish a *prima facie* case of obviousness. *See In re Mills*, 916 F.2d. 680 (Fed. Cir. 1990). Finally, the prior art references must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d. 981 (CCPA 1974); MPEP 2143.03. Specifically, "all words in a claim must be considered when judging the patentability of that claim against the prior art." *In re Wilson* 424 F.2d., 1382 (CCPA 1970). Using these standards, Applicants contend that the combination of Aihara and Beatty do not teach or suggest all features of the currently pending independent claims 8 and 16, some distinctive features of which are set forth in more detail below.

Aihara and Beatty, both alone and in combination, fail to provide teaching, suggestion or motivation for the claimed graphical user interface (GUI) window, which includes a breakpoint field that, upon receiving user input, performs the various functions set forth in claim 8. Independent claim 8 describes a software development tool and, particularly, a graphics rendering engine. The engine receives a first instruction address and produces a graphical user interface window. An example of such a window is set forth in Fig. 5 of the present specification. In particular, Fig. 5 illustrates a breakpoint field 66 that, upon receiving user input, selects a particular instruction address, such as address 0x1000 (Specification -- pg. 10, lines 11-27; Fig. 5). Thus, when a user highlights 67 a field within breakpoint area 66, the corresponding instruction address will be highlighted, that address being within a first sequence of instruction addresses (0x1000-0x1018) (Specification -- pg. 10, line 11 – pg. 11, line 27; Fig. 5). Specifically, the selected address is shown in particular stage of a processor pipeline. For example, the selected address 0x1000 is shown in the processor pipeline stage "EX," or execute stage (Specification -- pg. 9, line 4 – pg. 10, line 9; Figs. 3-5).

The Office Action alleges that Aihara discloses a graphical rendering engine coupled to receive the first instruction addresses and produce a graphical user interface. The Applicants disagree. As noted above, Aihara discloses screen shots (see, e.g., Figs. 13-15) that are only capable of <u>displaying</u> source code information. The screen shots disclosed by Aihara are <u>not</u> capable of <u>receiving</u> user input or responding to user input, and thus, are <u>not</u> equivalent to a graphical user interface (GUI), as presently claimed and known in the art.

In addition to failing to disclose a GUI, the Office Action concedes that Aihara does not teach that the screen shots may include "a breakpoint field [that] upon receiving user input via a pointing device selects a particular instruction address within the first sequence of instruction addresses shown in a particular stage of the pipeline" (Office Action, page 8). However, the Office Action alleges that Beatty uses breakpoint circuitry "to allow a user to 'establish at least one breakpoint for interrupting the operation of' the program" in column 3, lines 52-57 of Beatty (Office Action, page 8). In light of such teaching, the Office Action alleges that Beatty can be combined with Aihara to overcome the deficiencies therein. The Applicants disagree for at least the reasons set forth below.

As noted in the Office Action, Beatty uses breakpoint circuitry that allows a user to establish at least one breakpoint for interrupting the operation of the processor (see, Office Action, page 8 and Beatty -- col. 3, lines 52-57). In addition, Applicants concede that Beatty uses a graphical user interface, or GUI (Beatty -- Figs. 4-5), for displaying architectural layout 420 of the processor, source code 460/470 of the program being run on the processor, and the current state of registers 430 when a breakpoint (BP) is encountered. However, and as described in more detail below, Beatty fails to provide teaching, suggestion or motivation for a "graphical user interface" and "breakpoint field," as specifically recited in present claim 8.

First of all, Beatty fails to fully describe or illustrate the "breakpoint circuitry" mentioned in column 3, lines 52-27, and furthermore, fails to provide any manner by which the graphical user interface may be used to establish a breakpoint. As such, Beatty only provides vague teaching for adding a breakpoint to the program code (a practice that is extremely common to software debugging).

When a breakpoint is encountered, Beatty teaches that the user may visually inspect the contents of windows 420, 430, and 460/470 to determine if an error exists in the program code. If no errors exist, the user enters a continue command (cont) into the debug interface window 450. This allows processor operation to continue until the next breakpoint is encountered, at which time the state of register 430 is updated and user inspection of the windows proceeds (Beatty -- cols. 7-9; Figs. 4-5).

Although Beatty teaches that a breakpoint may be established by a user (i.e., added to the program code), Beatty does not teach or suggest that the GUI shown, e.g., in Figs. 4-5, includes a "breakpoint field" that is capable of receiving user input via a pointing device, as presently claimed. For example, Beatty provides absolutely no teaching or means by which a user can position the selection icon of a pointing device over, e.g., the breakpoint designated at line 16 of source code window 470 for selecting line 16 or performing any other functions set forth in claim 8. Instead of indicating user selection, Beatty teaches that the breakpoint at line 16 is highlighted (530, Fig. 5) merely to indicate to the user that program operation was interrupted at line 16 (Beatty -- col. 8, line 67 – col. 9, line 2; and Fig. 5).

If the teachings of Beatty were somehow combined with those of Aihara, the combination would enable a user to establish a breakpoint. The combination would also provide a graphical user interface by which a user may examine the operational states of the processor/program code that occur at the breakpoint. However, the combination would **not** enable a user to provide user input to a breakpoint field of the graphical user interface. In addition, the combination would **not** be capable of performing the functions recited in claim 8 (i.e., selecting a particular instruction address, displaying all instruction addresses, assigning a designator and assigning a non-designator) in response to user input received at a breakpoint field. As a consequence, the combination would fail to read upon all limitations of claim 8.

Aihara and Beatty, both alone and in combination, fail to provide teaching, suggestion or motivation for the claimed graphical user interface (GUI) window, which includes an instruction address field that, upon selection by a user via the pointing device, allows the user to move an instruction address. Claim 8 not only allows for rendering of a particular instruction address, but also allows a user to move an instruction address via a pointing device in order to optimize the flow of instructions through the processor pipeline. For example, a user can move instruction address 0x1000 to designate or highlight more execution stages within the first sequence of instructions in order to improve the processor throughput (Specification -- pg. 4, lines 12-16; pg. 4, line 28 – pg. 5, line 3; pg. 11, line 16 – pg. 12, line 10; pg. 16, line 27 – pg. 17, line 7; Fig. 5).

Neither Aihara nor Beatty, either singularly or in combination, mention an instruction address field that can be selected by a user pointing device, much less one that can be moved via a pointing device as presently claimed. The Office Action alleges that Aihara provides teaching for the claimed "instruction address field" on page 6, ¶ 0080. The Applicants disagree.

Although Aihara mentions a screen editing module 26 (¶ 0080; Fig. 17), Aihara specifically teaches that the screen editing module is used to "divide a frame into pluralities for displaying information obtained by the resource information module and the pipeline information module" (Aihara -- ¶ 0084; Fig. 17). However, dividing a frame or screen for purposes of displaying different types of information is **not** equivalent to moving an instruction address with a pointing device (i.e., for the purpose of optimizing the flow of instructions through the processor pipeline). As such, Applicants traverse the allegation that teaching or suggestion for the claimed "instruction address field" can be found within Aihara. In addition, Applicants assert that Beatty fails to provide teaching, suggestion or motivation for such a field, and thus, cannot be combined with Aihara to overcome the deficiencies therein.

Aihara and Beatty, both alone and in combination, fail to provide teaching, suggestion or motivation for the presently claimed method, which includes selecting a breakpoint within a breakpoint column of a display screen via user input at a breakpoint location on the display screen. Independent claim 16 has been amended to clarify that the breakpoint is selected via user input at a breakpoint location (e.g., highlighted area 67 of breakpoint field 66) on the display screen. As noted above, Aihara and Beatty each fail to provide teaching, suggestion or motivation for selecting a breakpoint field shown in a graphical user interface via user input with a pointing device. For at least the same reasons, Aihara and Beatty also fail to provide teaching, suggestion or motivation for selecting a breakpoint within a breakpoint column of a display screen via user input at a breakpoint location on the display screen. As a consequence, Aihara and Beatty cannot be relied upon to render all limitations of claim 16 obvious.

For at least the reasons set forth above, claims 1, 8, and 16 are believed patentably distinct over the cited art. Moreover, dependent claims 3-7, 9-15, and 17-20 are also believed patentably distinct for at least the same reasons are their respective base claim. Accordingly, removal of this rejection is respectfully requested.

CONCLUSION

The present amendment and response is believed to be a complete response to the issues raised in the Office Action mailed December 28, 2007. In view of the amendments and remarks herein, Applicants assert that pending claims 1 and 3-20 are in condition for allowance. If the Examiner has any questions, comments, or suggestions, the undersigned attorney earnestly requests a telephone conference.

No fees are required for filing this amendment; however, the Commissioner is authorized to charge any additional fees which may be required, or credit any overpayment, to LSI Logic Corporation deposit account no. 12-2252.

Respectfully submitted,

/Kevin L. Daffer/
Kevin L. Daffer
Reg. No. 34,146
Attorney for Applicant(s)

Daffer McDaniel, LLP
P.O. Box 684908
Austin, TX 78768-4908
(512) 476-1400
Date: April 28, 2008
JMF